

Realitat mixta per al mapat automàtic d'entorns de treball

Jhoan Estiben Granada Ruiz

Resum– En aquells entorns de treball en què s'hi mouen robots mòbils autònoms, com magatzems i fàbriques, cal que disposin d'un mapa de la zona per saber per on s'han de moure. Per generar aquests mapes hi ha moltes tècniques. Una d'elles és la que s'explora en aquest treball. La idea és tenir un robot explorador que vagi descobrint la localització de tots els obstacles. Aquest robot té una versió virtual que, cada cop que el real detecta un obstacle no present a l'entorn simulat, provoca la creació d'aquest obstacle al món virtual. D'aquesta manera, s'acaba creant un entorn virtual que acaba essent un mapa de localització dels obstacles i de les vies de pas de l'àrea de treball. Per fer-ho, el robot compta amb un mòdul sensor que fusiona les dades del lliadar real amb les del virtual per als controladors de més alt nivell del mateix robot per actualitzar l'entorn (creació o modificació d'objectes). Addicionalment, el mateix mòdul pot servir per fer que el robot treballi en mode de "realitat augmentada", tot detectant objectes que només són presents a l'entorn virtual. Els resultats demostren que s'ha desenvolupat un mòdul controlador capaç de realitzar les funcionalitats proposades. I el treball realitzat pot ser el punt de partida per a futurs estudis.

Paraules clau– Realitat Mixta, Robot explorador, Fusió de sensors, Sincronització de robots.

Abstract– In those work environments where autonomous mobile robots move, such as warehouses and factories, they need to have a map of the area to know where they move. There are many techniques for generating these maps. One of them is the one explored in this paper. The idea is to have a scout robot that will discover the location of all obstacles. This robot has a virtual version that, every time the real one detects an obstacle not present in the simulated environment, causes the creation of this obstacle in the virtual world. In this way, a virtual environment is created, which ends up being a map of the location of obstacles and passageways in the work area. To do this, the robot has a sensor module that merges real lidar data with virtual data for the highest-level controllers of the same robot to update the environment (creation or modification of objects). Additionally, the same module can be used to make the robot work in "augmented reality" mode, detecting objects that are only present in the virtual environment. The results show that a controller module capable of performing the proposed functionalities has been developed. And the work done can be the starting point for future studies.

Keywords– Mixed Reality, Explorer Robot, Sensor fusion, Robot synchronization

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

L A idea de realitat mixta (MR) ja és present a diverses indústries, fàbriques o magatzems, a causa dels diversos beneficis que pot tenir un mapat de l'àrea de treball per planificar rutes, o bé per valorar si la disposició

dels elements dins de l'entorn és l'adient, o per avaluar l'efecte que tindria modificar l'entorn, simplement fent canvis en el món virtual. La definició original de realitat mixta (1994), expressa que realitat augmentada (AR) i virtualitat augmentada (AV), són dues instàncies de MR [1]. En AR, els objectes virtuals es projecten en un entorn físic, mentre que en AV, els objectes físics són incorporats a un entorn virtual. En aquest treball es pretén assolir realitat virtual quan s'incorporin objectes reals (generalment obstacles) al món virtual. I d'altra banda, realitat augmentada, en el cas de fer servir dades virtuals (un possible obstacle virtual no present a la realitat) que afectin les decisions del robot real.

- E-mail de contacte: jhoanestiben.granada@e-campus.uab.cat
- Menció realitzada: Enginyeria de Computadors
- Treball tutoritzat per: Lluís Ribas Xirgo (Microelectrònica i Sistemes Electrònics)
- Curs 2020/21

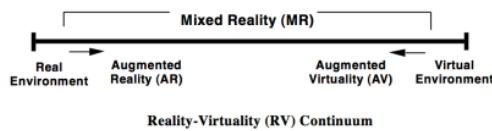


Fig. 1: Realitat-Virtualitat de [1]

L'objectiu del projecte és tenir un entorn de simulació amb un robot mòbil que tingui una contrapartida real. Tots dos segueixen les mateixes instruccions de moviment mentre adquireixen dades dels seus entorns. Les dades del sensor real es combinen amb les dades virtuals de manera que quan no s'identifiquen objectes reals a l'entorn virtual se'n crea una versió simulada i s'utilitzen les diferències entre objectes reals i virtuals per actualitzar la ubicació del robot real i/o les característiques dels objectes virtuals. A més, hi pot haver objectes virtuals purs que augmentin la complexitat dels entorns, per exemple, per a l'anàlisi del seu efecte en la realitat. El problema crític a resoldre és la combinació entre les dades del sensor real i les del virtual.

2 ESTAT DE L'ART

Com hem esmentat anteriorment, el tema tractat és bastant útil, per tant diversos articles tracten temes relacionats amb el desenvolupat. En un primer cas [2] es presenta la possibilitat d'incorporar un model basat en agents (ABM) per permetre millorar el procés de transport de sistemes basats en robots, principalment robots autònoms guiats (AGV) en diferents entorns de treball com poden ser indústries, fàbriques o magatzems. El model proposat presenta dos agents diferenciats, un per controlar els robots i un altre per tots els altres elements incloent-hi les ordres de transport. En aquest cas el concepte interessant és mantenir a la vegada un entorn simulat de la planta actual on s'està treballant, per tal d'optimitzar la presa de decisions dels robots i millorar les rutes. Aquest procés presenta un cert grau de dificultat, que comporta el fet de sincronitzar dades rebudes de l'entorn virtual i real, per això es presenta un mecanisme de sincronització per tal d'assolir la problemàtica esmentada.

Un segon exemple [3], presenta una simulació en temps real que permet obtenir la posició i orientació d'un robot basat en el sistema operatiu del robot (ROS) [4]. El software de la simulació és V-REP (actualment CoppeliaSim). El simulador rep l'orientació mitjançant un IMU i la posició actual a partir del paquet ROS, això permet mostrar l'orientació i la posició del robot en temps real. Un dels principals objectius és aconseguir facilitar els processos de teleoperació per tal de donar suport eficient i acurat a la navegació en l'entorn remot, i proveir l'operador amb un sentit de presència a l'entorn. Convencionalment es fa servir una càmera incorporada al robot perquè el teleoperador pugui observar en temps real el robot. En aquest cas concret com a sistema operatiu del robot es fa servir ROSARIA. Aquest operatiu rep comandes introduïdes per teclat des d'un ordinador teleoperador. Seguidament, l'ordinador teleoperador es comunica amb un ordinador controlador mitjançant wifi. I finalment l'ordinador controlador es comunica amb el robot per Bluetooth. El sistema ROS permet que els no-

des publiquin informació i d'altres es puguin subscriure i obtenir-la. En aquest cas el sensor IMU es comunica amb el sistema operatiu del robot, i envia dades. Les diferents dades rebudes per ROS es filtren i es publiquen perquè el simulador les pugui rebre. L'article presenta diversos experiments: Un d'ells consisteix a canviar l'angle de l'IMU i veure com varia en la simulació. Un segon experiment consisteix a comprovar el moviment del robot real i el simulat alhora. Els resultats demostren que no hi ha gaires diferències entre els valors de la simulació i els reals.

Un altre exemple [5] és un sistema que fa servir les ulleres Microsoft HoloLens per permetre la immersió en MR. L'objectiu principal és aconseguir un sistema que permeti programar robots per operadors amb poc coneixement en programació. Per aconseguir-ho es dissenya un sistema que incorpora un entorn virtual simulat, i permet als operadors donar ordres als robots virtuals, veure els resultats de la simulació, i després decidir si volen aplicar aquest moviment als robots reals amb diversos procediments que permeten en alguns casos optimitzar les trajectòries.

3 EXPLICACIÓ DE L'ENTORN

La tecnologia base és l'entorn CoppeliaSim amb el model del robot de la UAB per a Sistemes encastats. Aquest és un dels simuladors més utilitzats per recerca i educació [3]. Típicament es fa servir en el disseny i test d'algorismes de control per reduir el cost d'incorporar prototips físics. CoppeliaSim inclou un entorn de desenvolupament basat en una arquitectura de control. Per això cada objecte modelat pot ser controlat via un script encastat, un plugin, un node ROS, una API de client remota, o bé una solució personalitzada. CoppeliaSim pot operar de manera independent o bé pot ser incorporada en l'aplicació client principal.

3.1 Descripció del sistema original

Com s'ha esmentat l'objectiu principal és mantenir dues instàncies, una real i una altra virtual. En les següents subseccions presentarem els elements del sistema i ens centrarem a explicar l'Arquitectura Software inicial sobre la qual es treballarà.

3.1.1 Robot virtual

El robot virtual inclou diversos components per tal de simular el comportament del robot real. Per modelitzar el cos del robot, l'Arduino Nano i el mòdul d'alimentació del robot fem servir cuboides. El robot virtual presenta drivers que permeten simular el comportament del robot real i un controlador que regeix el comportament. El controlador del robot s'implementa mitjançant una màquina d'estats en Lua. La raó principal de fer servir màquines d'estats és perquè per la naturalesa reactiva permeten modelitzar el comportament dels controladors, a més faciliten la verificació del comportament del sistema, i la síntesi cap al SW és simple. D'altra banda, la raó de fer servir Lua és perquè els scripts desenvolupats al simulador que es fa servir es realitzen en aquest llenguatge. Continuant amb l'arquitectura del robot, aquest té dues rodes modelades amb cilindres que permeten el moviment del robot, cada roda és impulsada per un servomotor continu, modelat com un revolte joint. Té una

petita roda (roda boja) sobre la qual descansa la part posterior del robot, i li permet girar 360°. I compta amb un sensor de força per ajustar-se al cos del robot. Finalment, un sonar amb sensor ultrasònic i un servomotor estàndard (no continu) que controla l'orientació del sonar.

3.1.2 Robot real

El robot real (NinjUAB) incorpora tots els elements del robot virtual esmentats anteriorment més alguns altres que es van ometre en el model per simplicitat. Tots els elements es troben en una plataforma metàl·lica o de plàstic dur, subjectats directa o indirectament. Presenta les mateixes tres rodes que el robot virtual, a més presenta un mòdul d'alimentació de 5 piles (1,2V), on el pas de l'energia es controla mitjançant un interruptor. El robot presenta un microcontrolador Arduino Nano. Finalment, diferents drivers i un controlador programat en Sketch que regeix el comportament del robot.

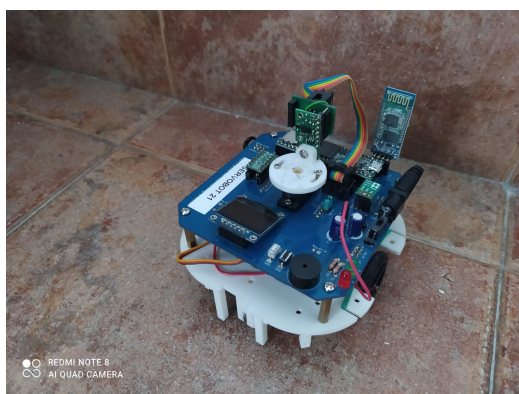


Fig. 2: Robot real

3.1.3 Arquitectura Software

Per a desenvolupar el projecte es fa servir el simulador de CoppeliaSim i algunes de les funcionalitats que inclou. En aquest simulador s'ha creat una escena i s'ha afegit alguns objectes per poder realitzar el mapat. Com per exemple una superfície, cubs (obstacles), el robot virtual i d'altres elements per permetre la comunicació amb el robot real. Aquests objectes presenten scripts associats, programats en Lua que permeten modelar el seu comportament. Els scripts a desenvolupar es caracteritzen per tenir una estructura definida. Bàsicament consten de quatre parts encapsulades en funcions. La primera (initialitzation) i l'última (cleanup), s'executen una vegada, abans d'iniciar la simulació i en finalitzar-la. Per contra, la segona part (actuation) i la tercera (sensing), ho fan cada pas de la simulació i, per tant, múltiples vegades. El software desenvolupat a CoppeliaSim sobre el qual es treballarà compta amb diferents nivells de controladors. Un conjunt de nivells per tal de permetre les diferents funcionalitats. Tals com l'enviament correcte de comandes als robots, la recepció de notificacions per part dels robots (detecció d'obstacles). I altres una mica més especialitzades com seguir rutes. Esmentar que el funcionament dels diferents nivells és paral·lel, i la simulació es realitza en temps real. Els diferents controladors es modelitzen mitjançant diferents màquines d'estats finits per les raons esmentades prèviament (modelització, síntesi

i verificació). En un primer lloc trobem el nivell L0, que en ambdós robots és el mateix, però un programat en Lua (robot virtual) i l'altre en Sketch (robot real). El nivell L0 és l'encarregat d'executar ordres bàsiques. Un nivell per sobre trobem el controlador *L0L1_Link* encarregat d'actuar d'enllaç entre el L0 i L1. La seva finalitat és aconseguir la sincronització entre el *L0_virtual* i el *L0_real*. Aquest controlador és on se centrarà el desenvolupament del projecte. Per sobre del nivell de sincronització anterior trobem el L1. Aquest simplement actua com a seguidor de camins. Pot funcionar de manera manual rebent comandes a través d'una interfície, o bé de manera automatitzada.

A continuació els diferents missatges que s'intercanvien entre els diferents nivells de controladors.

L1 es comunica amb *L0L1_Link* enviant comandes bàsiques esmentades prèviament, resumides en la taula següent.

TAULA 1: ORDRES BÀSIQUES L1

Comanda	Acció
GO	Rotació i desplaçament.
SONAR	Inici de l'escaneig.
RESUME	Continuació de l'escaneig.
HALT	Avortar la missió actual.

Continuant en la mateixa línia, el nivell *L0L1_Link* envia a L1 diferents missatges referents a si s'ha pogut realitzar correctament la comanda ordenada per L1 o no.

- D: RAY α distància: resultat de realitzar un sondeig amb el sonar.
- W:BLOCKED distància: resultat d'haver trobat un obstacle en el camí, retornem la distància recorreguda.
- Error_sincronització: no s'ha pogut realitzar la sincronització.
- D: GO OK: el moviment s'ha pogut completar correctament

El nivell *L0L1_Link* a la vegada, es comunica amb L0 (virtual i real) i li envia les ordres bàsiques que ha rebut prèviament de L0. I rep respostes dels robots, que poden ser diferents tipus de missatges. A continuació els més rellevants (a la secció A2 de l'apèndix es pot observar una taula amb tots els missatges).

- D: GO OK: el moviment s'ha pogut completar correctament
- D: RAY α distància: resultat de realitzar un sondeig amb el sonar.
- W:BLOCKED distància: com ha resultat d'haver trobat un obstacle en el camí, retornem la distància recorreguda.
- D: GO OK: el moviment s'ha pogut completar correctament
- D: SONAR OK: l'operació sonar ha finalitzat correctament

Per finalitzar un resum gràfic de l'arquitectura software:

- L0: nivell d'ordres bàsiques.
- LOL1_link: nivell que permet enviar comandes provinents de L1 els robots i sincronitzar les respostes.
- L1: nivell seguidor de camins, que permet connectar el nivell L0 i un possible nivell L2(planificador). Rep conjunt de comandes i envia comandes individuals.

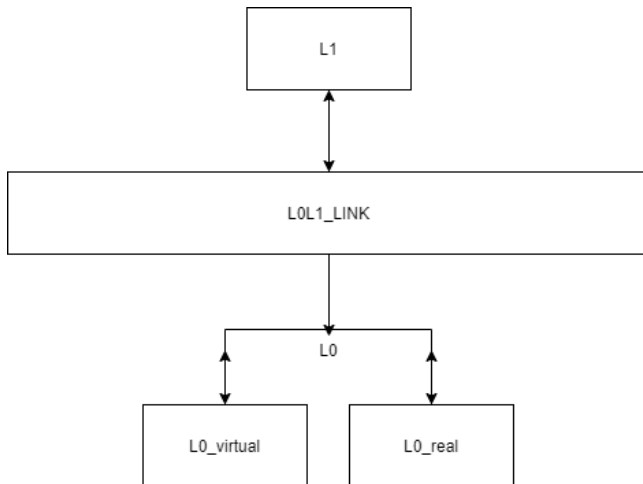


Fig. 3: Arquitectura Software

4 CONTROLADOR DE SINCRONITZACIÓ (LOL1_LINK)

Com hem vist anteriorment en la secció 2, la sincronització de dades en entorns reals que tenen contrapartides virtuals, és una qüestió bastant tractada i clau en molts sistemes. El fet de tenir constància que la resposta que es rep fa esment a l'última ordre donada, i en cap cas estem barrejant dades, és molt important. Per això mateix, el treball està centrat a solucionar aquesta problemàtica. En aquest cas, s'ha decidit realitzar un disseny del controlador per fases per tal de simplificar el seu disseny. Es proposa un primer model que permeti solucionar només la sincronització i fusió a nivell comanda SONAR i RESUME. Per tant s'espera poder realitzar un mapat de l'entorn estàticament, exclusivament rotant el sonar incorporat en ambdós robots.

Per realitzar la implementació del controlador de sincronització inicial, s'ha realitzat una màquina d'estats Mealy que compta amb 4 estats:

- SYNC: estat inicial on s'esperen respostes, si aquestes arriben al mateix temps ens mantindrem aquí.
- BEHIND: estat al qual passarem en cas de rebre una resposta del robot real abans del robot virtual. Si rebem resposta compatible abans del timeout tornarem a SYNC.
- AHEAD: estat on esperarem una resposta endarrerida del robot real en haver rebut una del virtual. Si rebem resposta compatible abans del timeout tornarem a SYNC.

- NO_SYNC: anirem aquest estat quan no s'hagi pogut dur a terme la sincronització.

El primer funcionament general es podria resumir següentment com:

1. S'envia una comanda SONAR o RESUME.
2. S'espera per la resposta dels robots.
3. Si rebem resposta dels dos a la vegada, avaluem la resposta i l'enviem cap al nivell superior(L1).
4. En cas de rebre resposta d'un dels dos, esperem un cert temps (timeout) la resposta de l'altre.
5. En cas de rebre la resposta del segon en un temps inferior al timeout, avaluem la resposta (6). Si la resposta no es rep passat el timeout, enviem un missatge d'error informant que la sincronització no ha estat possible.
6. En avaluar la resposta poden passar diferents situacions:
 - Les respostes són compatibles, és a dir, fan referència a la mateixa comanda, aleshores les podem fusionar.
 - Les respostes són incompatibles, no fan referència a la mateixa comanda i no les podem fusionar. Tornaríem a 4.
7. Si les respostes són compatibles, passem a la fase de fusió.
 - Si no es detecta cap objecte en el món real, o bé si es detecta en ambdós, es continua normalment i es torna a 1.
 - Si es detecta objecte en el món real a una certa distància determinada, i no es detecta en el món virtual, es procedeix a incorporar l'objecte al món virtual i es torna a 1.

La comprovació de compatibilitat entre comandes es realitza bàsicament comprovant que l'angle de les respostes dels robots sigui el mateix. Recordar que les respostes a les comandes SONAR i RESUME tenen el format següent: "D: RAY α dist".

També esmentar que el nivell L0 en rebre aquest es desplaça un cert angle i retorna la distància a l'objecte més proper. Concretament es desplaça als següents angles:

Angles = {15, 30, 45, 60, 75, 90, 40, 0, -15, -30, -45, -60, -75, -90, -40}

Per tant no hi ha possibilitat que dues respostes es puguin confondre en comandes diferents. Tot seguit, pel que fa a la fusió de dades simplement es crea un objecte virtual si aquest és present en el món real i no hi és en el món virtual. És a dir, si rebem com a respostes:

- L0_virtual: "D:RAY α 0"
- L0_real: "D:RAY α dist"

Finalment en el cas d'haver de dibuixar un objecte. Procedirem a obtenir la posició del sonar virtual. A partir de la posició, l'angle actual de rotació i la distància a l'objecte real, dibuixem l'objecte.

El càlcul del punt en la simulació on s'ha de situar l'objecte es realitza de la següent manera. Donat que coneixem la distància i els angles respectius podem formar un triangle i calcular les posicions X i Y corresponents.

$$X = posRobot + \cos(\alpha) * dist2objecte \quad (1)$$

$$Y = posRrobot + \sin(\alpha) * dist2objecte \quad (2)$$

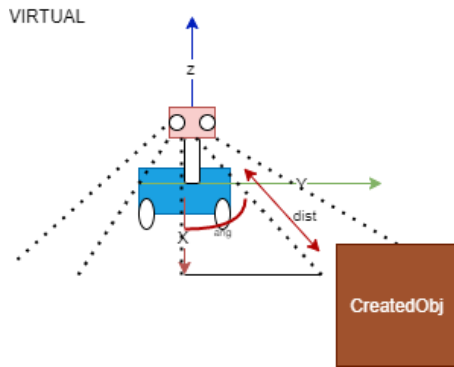


Fig. 4: Creació d'objectes al món virtual

Tot seguit es presenta un segon model una mica més complex tenint en compte la comanda GO. Aquesta comanda pot rebre dues respostes diferents per part dels robots:

- “GO OK”: la comanda GO ha estat completada correctament.
- “W:BLOCKED dist”: la comanda no s’ha pogut completar correctament i resten “dist” (cms) del trajecte total per ser completat.

A causa d’això apareixen diverses situacions possibles que hem de solucionar:

- Ambdós retornen “GO OK” alhora. En aquest cas, enviem la resposta cap a L1.
- Un robot retorna “W:BLOCKED dist” i l’altre no retorna res. En aquest cas haurem d’esperar un cert temps per veure si la contrapartida troba l’obstacle. Si l’obstacle és detectat en la realitat, i no existeix en el món virtual, procedirem a dibuixar l’obstacle. I continuarem la simulació fins que el robot virtual trobi aquest obstacle. Si el troba, ja podríem retornar el mateix missatge al nivell L1 (tots dos estan sincronitzats).
- En el cas que un robot retorna “GO OK” i l’altre retorna “W:BLOCKED dist”, tot i que hauria de donar-se una diferència de desplaçament bastant gran entre els robots. Si el robot real ha finalitzat correctament, i el virtual s’ha bloquejat, simplement enviem el missatge real L1. En l’altre cas haurem de dibuixar un obstacle. En aquest punt haurem de verificar que el robot i el nou objecte no estiguin molt a prop i potser seria més adient situar el robot virtual a la mateixa distància que es troba el robot real de l’objecte. També haurem de tenir en compte que els objectes virtuals que puguin haver-hi presents en l’escena anteriorment no estiguin molt a

prop de l’objecte a dibuixar, en aquest cas podríem optar per no dibuixar cap obstacle o veure alguna manera de fusionar dos objectes virtuals.

Finalment es realitza un tercer disseny que té en compte d’una manera més acurada els bloquejos en trobar obstacles per part del robot real. Resumidament:

- 1. Partint del estat “SYNC” en rebre un “GO OK” virtual aniríem cap a “AHEAD”. En exhaurir el timeout, anirem a un nou estat de possible reposicionament “RE_POS”
- 2. En aquest nou estat “RE_POS”, esperarem un “W:BLOCKED dist” per part del robot real o bé un “GO OK”.
- 3. En cas de rebre un “W:BLOCKED dist” fem servir aquesta informació per reposicionar el robot virtual i dibuixar el nou objecte i ens dirigim a “SYNC”. Si rebem un “GO OK” simplement anem a “SYNC”.

Si a l’estat “SYNC” no rebem cap “GO OK” virtual, però rebem un “W:BLOCKED dist” per part del real, haurem de tenir en compte que el robot virtual s’haurà desplaçat (mentre el real ha estat parat) i no estarà sincronitzat respecte al real. Per tant l’haurem de reposicionar. A més d’aquesta funcionalitat, s’afegeix la incorporació d’obstacles que es troben a una certa distància en el món virtual diferent del món real (abans només s’incorporava si l’obstacle no era present al món virtual), i s’elimina l’estat “NO_SYNC” perquè realitzava les mateixes funcionalitats que l’estat “SYNC”

A la figura(5) podem observar aquest nou disseny en forma de FSM.

5 EXPERIMENTS I RESULTATS

En aquesta secció consten els experiments dissenyats per tal de comprovar el correcte funcionament de les diferents versions del mòdul sensor agregat i els resultats. Els resultats es poden visualitzar en el següent [7].

5.1 Primer experiment

Per validar el correcte funcionament del primer controlador de sincronització hem dissenyat un senzill experiment. Partint d’una simulació buida (sense obstacles). Preparem un conjunt d’objectes reals de (11.5 x 11.5 cm) i comprovem com en detectar un objecte real a una distància de 60 cm, no present en el món virtual aquest és afegit. Els obstacles són situats paral·lelament, separats entre ells a una distància de 25 cm. L’obstacle central es troba a 50 cm del robot. En aquest experiment farem servir la comanda SONAR i un conjunt de comandes RESUME per tal de realitzar l’escaneig. Les comandes seran introduïdes manualment des de la interfície d’usuari (UI) del controlador L1. En realitzar passades successives comprovarem que els objectes són detectats en ambdós mons a una distància aproximadament igual.

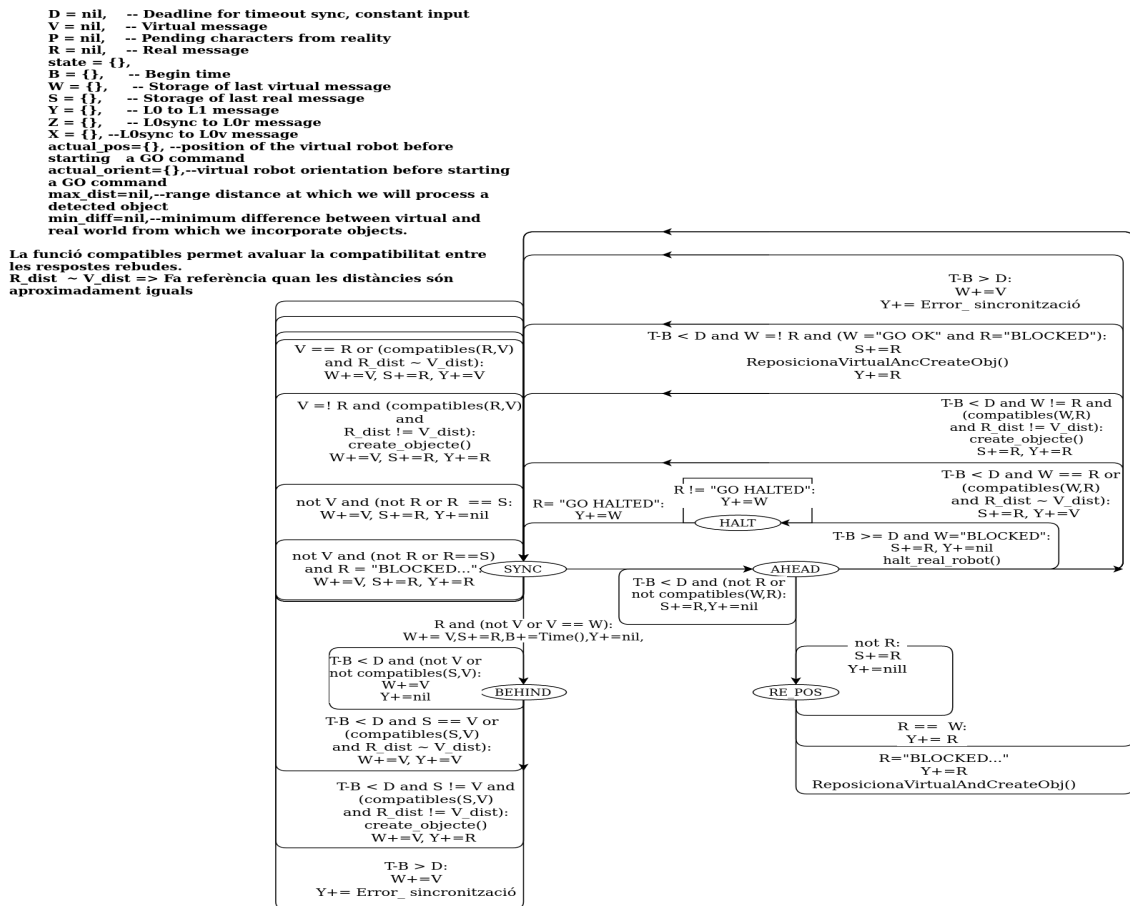


Fig. 5: Disseny final LOL1.LINK FSM

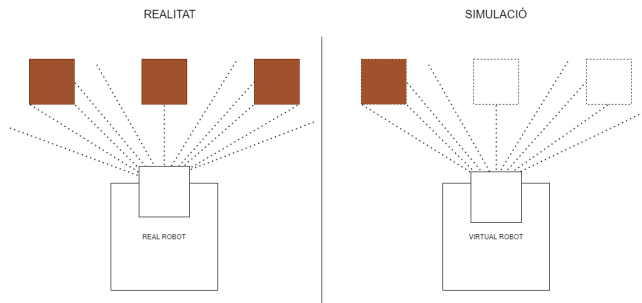


Fig. 6: Primer experiment

5.1.1 Resultats experiment 1

Els resultats han estat els següents: els objectes eren detectats i s'incorporaven en el món virtual. Però cada vegada que es detectava un obstacle a un cert angle, l'objecte era incorporat a la simulació provocant en alguns casos col·lisions amb altres objectes, o incorporacions de més d'un obstacle. La raó d'això, és que certs objectes massa grans són detectats en diferents angles per part del robot real. Quan incorporem l'objecte detectat a un cert angle en el món virtual, el centrem respecte a aquest angle, en la realitat, potser, aquest només ha estat detectat per uns pocs centímetres en aquell angle. Per tant en incorporar-lo al món virtual, la simulació presentarà diferències amb el món real provocant un funcionament erroni en general. També pot influir una característica d'implementació del sonar virtual, el qual per reduir temps, una vegada ha completat una comanda RE-

SUME a un cert angle, es dirigeix cap al següent. Això pot provocar que l'objecte que ha de ser incorporat no sigui detectat justament després d'afegir-lo (en passades posteriors sí). Una primera alternativa és controlar les interseccions amb els objectes presents a la simulació abans d'afegir un de nou. Una altra podria ser, realitzar un escaneig més elaborat on no s'afegeixi l'objecte en el primer angle detectat. Seguidament es prepara un escenari semblant, on els objectes només són detectats a un cert angle i en aquesta situació els objectes són incorporats correctament i detectats en passades successives en ambdós mons.

5.2 Segon experiment

Seguidament, per validar una part del funcionament del segon disseny del mòdul sensor agregat, es prepara novament una escena virtual buida. S'espera realitzar un recorregut en el qual el robot real es trobi un objecte, i aquest sigui incorporat al món virtual, i després l'objecte afegit provoquei una aturada del robot virtual davant d'aquest objecte. Es realitzarà fent servir una única comanda GO introduïda per la UI. En la següent figura podem observar el descrit anteriorment.

5.2.1 Resultats experiment 2

Els resultats en aquest punt han estat els següents: el robot real en detectar un obstacle triga un cert temps a donar l'avís, això provoca que el robot virtual segueixi el trajecte sense possibilitat de dibuixar l'objecte davant per aturar-lo.

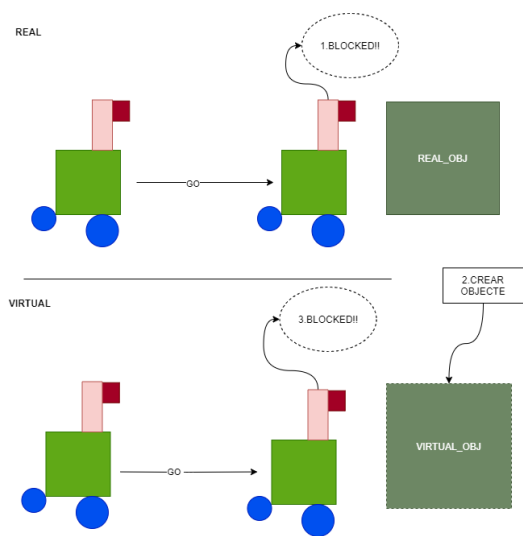


Fig. 7: Segon experiment

Per solucionar-ho es realitza un tercer disseny del sensor agregat, que permeti tornar a situar el robot virtual a una posició aproximadament la mateixa que el robot real i dibuixar l'objecte. En aquest cas la implementació del tercer disseny té el resultat esperat.

5.3 Tercer experiment

Per validar una segona part del funcionament del segon disseny, es prepararà una escena virtual amb un objecte de (5 cm x 5 cm x 25 cm), per tal que el robot virtual el trobi i provoqui una aturada en el robot real. Es realitzarà fent servir una única comanda GO introduïda per la UI.

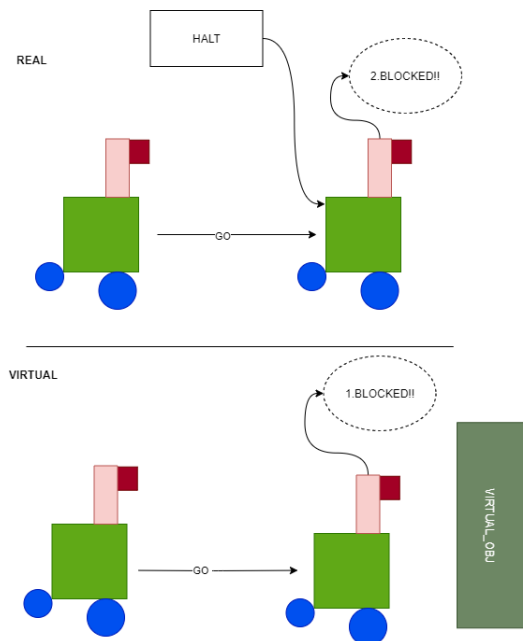


Fig. 8: Tercer experiment

5.3.1 Resultats experiment 3

El tercer experiment havia de validar que en detectar un objecte virtual es provoqui una aturada del robot real. En

aquest punt el funcionament ha estat l'esperat, només realitzar un comentari, el timeout en aquest punt pren més importància. Si esperem molt temps, trigarem molt a aturar el robot virtual, i aquest continuarà el camí realitzant més recorregut. Si esperem poc, aturarem el robot real abans de realitzar el mateix recorregut que el robot virtual.

5.4 Quart experiment

Es planteja un quart experiment semblant a l'experiment dos però amb major complexitat. El que es pretén és realitzar un recorregut fent servir comandes GO per tal d'incorporar diferents objectes reals a la simulació virtual, per mapar el terreny.

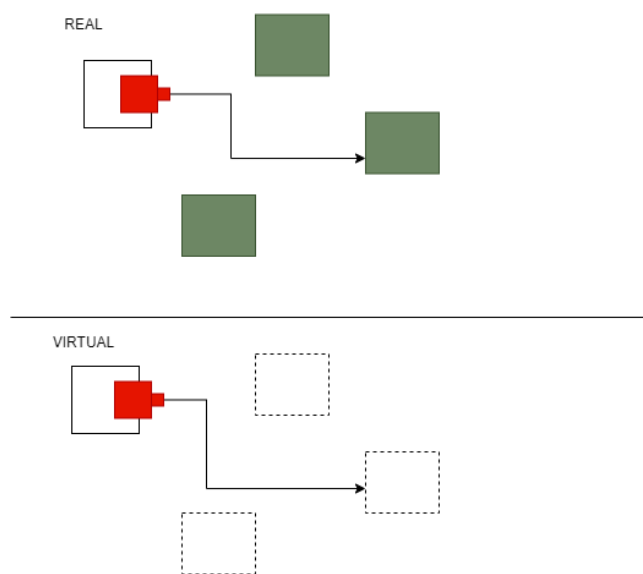


Fig. 9: Tercer experiment

5.4.1 Resultats experiment 4

L'experiment 4 a diferència del 2 incorpora rotacions en ambdós robots, i més d'una ordre, el que anomenaríem recorregut. Els objectes són incorporats correctament, per tant es pot arribar a fer un mapat d'una zona determinada.

6 CONCLUSIONS

L'objectiu principal del treball consistia a realitzar un mapat de l'entorn de treball fent servir un robot real i la seva contrapartida virtual. A partir d'aquí han sorgit diversos objectius parcials: Com la creació d'un entorn que permeti aquesta realitat mixta, permetent l'enviament i la recepció de dades dels robots; la sincronització les diferents respostes dels robots, la fusió de les diferents dades, el fet d'actualitzar el món virtual quan era necessari, i la presa de decisions en el món virtual a causa d'esdeveniments del món real i viceversa. Partint d'un entorn que permetia assolir realitat mixta. Els resultats demostren que s'ha desenvolupat un mòdul controlador capaç d'assolir les diferents funcionalitats proposades inicialment. Certament hi ha certes limitacions derivades de les diferències de funcionament entre els robots, la precisió de les dades o l'abast de la complexitat del disseny. Un exemple pot ser la part d'escaneig de l'entorn i la incorporació d'obstacles. És per això que el

treball realitzat, obre la possibilitat a ser millorat en diversos aspectes. Un d'ells podria ser realitzar un escaneig dels objectes més precís. Un segon aspecte, que queda fora de l'abast del treball podria ser una re-programació del robot real per adaptar-lo de millor manera als missatges i al comportament del robot virtual. Un tercer aspecte podria ser ajustar amb major precisió l'addició d'obstacles per assegurar que aquests són afegits a la mateixa distància, perquè ambdós mons presentin les menors diferències possibles, i d'aquesta manera la fusió de dades sigui més acurada.

AGRAÏMENTS

Agrair a tots els professors de l'escola d'Enginyeria, especialment als de la Menció de Computadors. I evidentment al tutor Lluís Ribas Xirgo, pel seu compromís i dedicació. A la meua família que m'ha donat el seu suport en el desenvolupament d'aquest projecte. Els meus amics, especialment al Khalid i l'Edgard per escoltar-me parlar durant moltes hores del treball. I com no, els meus dos gatets Mía i Tiger que m'han acompanyat en moltes hores a l'habitació.

REFERÈNCIES

- [1] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Proceedings of Tele manipulator and Telepresence Technologies*, 2351(34):282–292, 1994.
- [2] Chaile, I., Ribas-Xirgo, Ll. (2016). Running Agent-based-models Simulations Synchronized with Reality to Control Transport Systems. *Automatika*. 57. 452-465. DOI: 10.7305/automatika.60-1.1243.
- [3] N., Pinrath, N., Matsuhira. 2019. Real-time Simulation System for Teleoperated Mobile Robots using V-REP. In *Proceedings of the Advances in Robotics 2019 (AIR 2019)*. Association for Computing Machinery, New York, NY, USA, Article 5, 1–5. DOI:
- [4] ROS.org — About ROS. (2021). Retrieved 27 May 2021, from <https://www.ros.org/about-ros/>
- [5] M., Ostanin, A., Klimchik, 2018. Interactive Robot Programming Using Mixed Reality, *IFAC PapersOnLine*, Volume 51, Issue 22, Pages 50-55, ISSN 2405-8963, DOI: <https://doi.org/10.1016/j.ifacol.2018.11.517>.
- [6] E., Wilfried. An introduction to sensor fusion. Vienna University of Technology, Austria, 2002, 502: 1-28.
- [7] Recopilació experiments - TFG Mixed Reality (UAB). (2021). Retrieved 10 June 2021, from <https://youtu.be/L90w4wCprn>
- [8] CoppeliaSim User Manual. (2021). Retrieved 1 April 2021, from <https://coppeliarobotics.com/helpFiles/>

APÈNDIX

A.1 Planificació i Metodologia

El desenvolupament del projecte presenta un seguit de fases on les quals es desenvoluparan diferents tasques per tal d'assolir els objectius proposats. Per realitzar i organitzar aquest projecte s'ha decidit utilitzar la metodologia agile Kanban gràcies a l'eina Trello. Ja que per mitjà d'una taula amb tres columnes (To Do, Doing, Done) permet visualitzar tot el flux de treball.

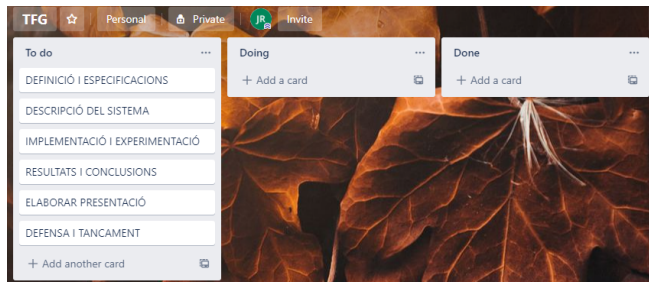


Fig. 10: Tercer experiment

- **DEFINICIÓ I ESPECIFICACIONS (14/03):** En aquesta fase s'ha realitzat la introducció i explicació de l'entorn de desenvolupament del projecte (inclosa la descripció de l'arquitectura software dels robots). S'ha dut a terme una proposta dels objectius principals del projecte, junt amb aquesta planificació. S'han incorporat referències bibliogràfiques i altra documentació relacionada amb el projecte. I finalment, s'ha efectuat el lliurament del robot real.
- **DESCRIPCIÓ DEL SISTEMA I IMPLEMENTACIÓ (25/04):** En aquesta fase s'ha realitzat la descripció (especificació) del model de comportament del sistema i, especialment, del mòdul sensor agregat que permet la sincronització de dades dels diferents robots. A més s'ha realitzat una primera implementació del primer disseny del mòdul sensor agregat.
- **IMPLEMENTACIÓ, EXPERIMENTACIÓ I INCLUSIÓ DE RESULTATS PARCIAIS (30/05):** En aquest punt a partir de l'especificació realitzada en la fase anterior, s'ha realitzat la programació, experimentació i obtenció i inclusió de resultats. Inicialment es plantejaven tres experiments:
 - Primer experiment: explorador estàtic que detecti objectes reals no presents en el món virtual i els incorpori. Escena virtual inicialment buida.
 - Segon experiment: seguidor de ruta, que té en compte objectes reals i virtuals, i incorpora els reals al món virtual. Escena virtual inicialment buida.
 - Tercer experiment: el primer però tenint en compte objectes virtuals presents a la simulació. Escena virtual amb objectes.

- **INCLUSIÓ DE RESULTATS I CONCLUSIONS (20/06):** En aquesta fase s'ha realitzat la inclusió dels resultats dels experiments a l'article, i la redacció de l'article final.

- **ELABORACIÓ DE PRESENTACIÓ (27/06):** La penúltima fase correspon a l'elaboració de la presentació.
- **DEFENSA I TANCAMENT (12/07):** Finalment s'ha de realitzar la defensa del treball.

A.1.1 Informe de Progrés I

En la secció següent es presentarà el informe de progrés I. En aquest informe constarà el nivell de seguiment de la planificació inicial, els canvis introduïts i les seves justificacions. A més es valoraran els possibles canvis introduïts en els objectius i en la metodologia, i les seves pertinents justificacions. Després s'explicarà la metodologia emprada per arribar a completar els objectius. I Finalment es presentarà la bibliografia addicional que s'ha emprat.

En primer lloc, la planificació ha estat modificada. S'esperava que en entregar aquest informe s'hagués realitzat una especificació i descripció del funcionament del sensor agregat. En aquest punt la descripció del sistema s'ha arribat a realitzar, però addicionalment per la part del mòdul sensor agregat, s'ha decidit realitzar la implementació d'una versió simplificada i el disseny de l'experiment que provi la validesa i el correcte funcionament d'aquesta part. La raó d'aquest canvi és purament per simplificar la complexitat del disseny i de la implementació. Una vegada correctament validat el seu funcionament s'espera continuar amb el 2n disseny més complex i la seva implementació. D'aquesta manera es considera que les revisions al model seran més senzilles.

En segon lloc, els objectius i la metodologia no han patit cap canvi respecte a l'informe anterior.

En tercer lloc, la metodologia emprada per aconseguir els objectius consisteix a dividir el treball en diverses tasques i buscar de dividir aquestes tasques el màxim possible, primer per reduir la seva complexitat i segon per facilitar la seva validació. En les següents imatges es pot veure el seguiment de les diferents fases i les diferents tasques acomplides en una fase determinada.

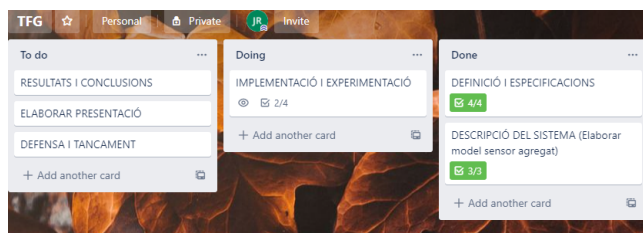


Fig. 11: Progrés TFG 25/04

La llista corresponent a la descripció del sistema és:

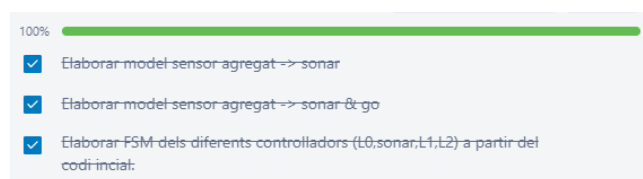


Fig. 12: Llista tasques de descripció del sistema

I seguidament la llista de tasques corresponent a la implementació i experimentació: Finalment en aquesta part



Fig. 13: Llista tasques d'implementació i experimentació

s'ha consultat sobretot la regular API de CoppeliaSim i la manipulació de strings de LUA:

- <https://coppeliarobotics.com/helpFiles/>
- <https://www.lua.org/manual/2.4/node22.html>

A.1.2 Informe de Progrés II

A l'informe de progrés II consta el nivell de seguiment de la planificació i els ajustos, una explicació general de la metodologia emprada, amb una explicació provisional dels resultats i les conclusions. Finalment la bibliografia emprada. La planificació ha estat lleugerament modificada. En aquesta fase s'esperava una implementació del segon disseny i experimentació en la qual es valoraven els resultats sense incloure'ls encara en el document final. En aquest punt la implementació ha estat realitzada, però sobre un tercer disseny (no gaire diferent del segon), i alguns dels resultats dels primers experiments efectuats ja estan inclosos, a causa del fet que es va considerar adient tenir una primera versió dels resultats. També s'ha inclòs una primera versió de les conclusions fins al moment. D'altra banda s'han realitzat algunes gravacions que permeten comprovar el funcionament del mòdul desenvolupat. S'han modificat algunes seccions com l'estat de l'art, s'han combinat experiments i resultats, s'han modificat algunes màquines d'estats de l'annex(L0,"sonar","sonar&go", i "sonar&go2") i s'han eliminat d'altres(sonar.controller,L1 i L2), ja que no presenten gaire importància per validar el funcionament del mòdul desenvolupat o bé són massa específiques, o no es fa servir tota la funcionalitat (per exemple L1, només es fa servir en mode manual) i L2 no es pensa fer servir. La metodologia que s'ha emprat ha estat la mateixa que en les fases anteriors. S'han identificat diverses tasques a realitzar i s'han anat realitzant per ordres de prioritat. En la següent imatge podem observar l'estat del projecte a l'inici de la fase:

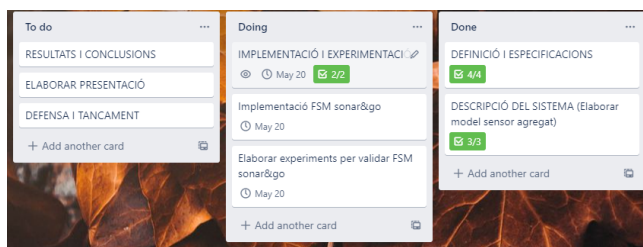


Fig. 14: Llista tasques d'implementació i experimentació

Provisionalment, els resultats obtinguts demostren que el disseny desenvolupat i la seva implementació és correcta. Tot i això, en fer escaneigs, certes variables s'han de tenir

en compte per fer un escaneig més realista i més funcionalitats haurien de ser incorporades, com la fusió d'objectes, per permetre que la mida i l'angle no afectin les diferències entre món real i virtual. D'altra banda pel que fa a l'altre tipus de comanda principalment els resultats són els esperats pel tercer disseny. Per concloure, podem afirmar que s'ha aconseguit desenvolupar un disseny que assoleix l'objectiu principal i els parcials derivats, però que pot incorporar encara més funcionalitats per permetre una generació d'un entorn més realista. Generalment hem vist que el desenvolupament del projecte no és lineal, i que en alguns casos, a causa dels resultats d'un experiment hem hagut de modificar el disseny i, la implementació. Per tant observem un procés cíclic. Finalment en aquesta part s'ha consultat sobretot els arxius d'ajuda de CoppeliaSim, concretament les propietats dels objectes i la regular API:

- <https://coppeliarobotics.com/helpFiles/en/commonPropertiesDialog>.
- <https://coppeliarobotics.com/helpFiles/>

A.2 L0: Missatges i Màquina d'estats

TAULA 2: RESPOSTES L0

Resposta	Causa
E:GO –	Error, angle erroni o sense angle
E: GO A -	Distància incorrecta o inexistent
W: GO 0 0	Distància i angle són zero
D: GO HALTed	Go avortat
D: GO OK	GO completat correctament
D:SONAR HALTed	Sonar avortat
D:RAY α dist	Resultat d'un ECHO. S'ha detectat un objecte en α a 'dist' cms
D: SONAR OK	Sonar completat correctament
D: SONAR RAY HALTed	Sonar Ray avortat
D: SONAR ECHO HALTed	Sonar Echo avortat
D: RESUME Halted	Resume avortat
W:BLOCKED dist	El robot ha trobat un obstacle, no pot continuar, i ha realitzat 'dist' cms de trajecte
D:BLOCKED HALTed	S'ha avortat mentre el robot estava bloquejat
W: Ignored command	Comanda ignorada. Pot donar-se quan s'està executant una comanda i al mateix temps es rep una altra.
E: Undefined state!	Error produït per un estat indefinit, provoca la parada del sistema.

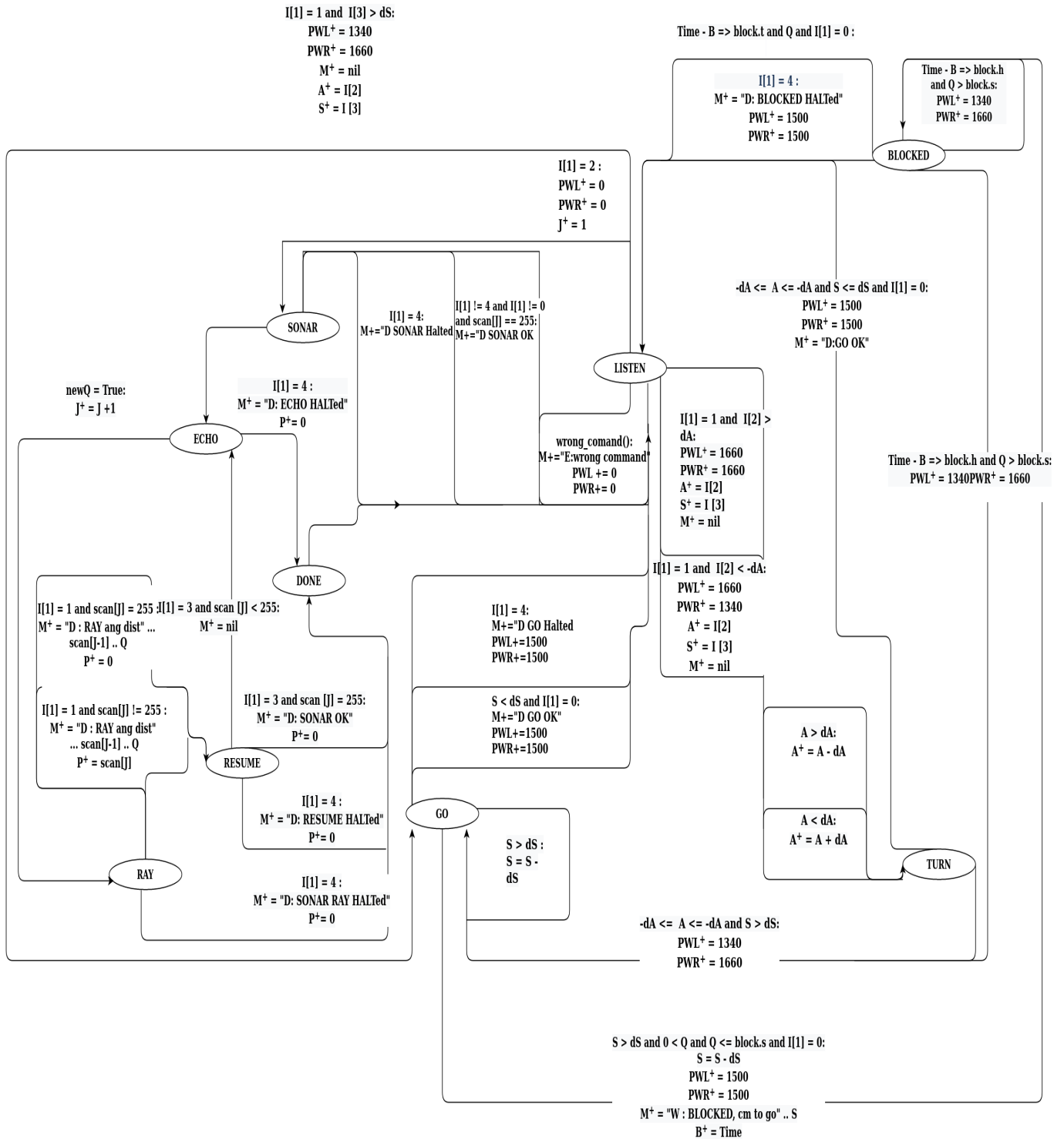


Fig. 15: FSM Controlador L0